

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Dokumentácia k inžinierskemu dielu

Bc. Michal Cihák

Bc. Michal Gajdoš

Bc. Lukáš Masár

Bc. Pavol Michálek

Bc. Vladimír Osvald

Bc. Matúš Pikuliak

Bc. Tomáš Sýkora

Tím (č. 12): Cats Can Code

Predmet: Tímový projekt I

Ročník: 1.

Akademický rok: 2014/2015

Vedúci práce: Ing. Lukáš Turský

Obsah

1	Úvod.....	4
1.1	Podiel členov tímu na dokumentácií.....	4
2	Ciele projektu pre zimný semester	5
3	Cel'kový pohľad na systém.....	5
3.1	Technologický základ a architektúra.....	5
3.2	Dátový model.....	6
3.3	Moduly.....	7
3.4	Referencie a dokumenty	7
4	Moduly	7
4.1	Zdieľanie stretnutia (čiastočne implementované).....	7
4.1.1	Analýza.....	7
4.1.2	Návrh.....	8
4.1.3	Implementácia.....	8
4.2	Generovanie odkazov (implementované).....	8
4.2.1	Analýza.....	8
4.2.2	Návrh.....	8
4.2.3	Implementácia.....	9
4.3	Vytváranie stretnutí (implementované).....	9
4.3.1	Analýza a návrh.....	9
4.3.2	Implementácia.....	9
4.4	Časy stretnutí (implementované).....	9
4.4.1	Analýza.....	9
4.4.2	Návrh.....	10
4.4.3	Implementácia.....	10
4.5	Miesta stretnutí (implementované).....	11
4.5.1	Analýza.....	11
4.5.2	Návrh.....	11
4.5.3	Implementácia.....	11
4.6	Hlasovanie (čiastočne implementované).....	13
4.6.1	Analýza.....	13

4.6.2	Návrh.....	13
4.6.3	Implementácia.....	13
4.7	Používatelia (čiastočne implementované).....	14
4.7.1	Analýza.....	14
4.7.2	Návrh.....	14
4.7.3	Implementácia.....	15
4.8	Notifikácie (neimplementované).....	16
4.8.1	Analýza.....	16
4.8.2	Návrh.....	16
4.9	Módy (neimplementované).....	16
4.9.1	Analýza.....	16
4.9.2	Návrh.....	16

1 Úvod

Predkladaný dokument slúži ako technická dokumentácia k našemu projektu. V tomto dokumente je zaznamenaná technická analýza, návrh, implementácia a testovanie projektu a jeho častí ako inžinierskeho diela. Cieľom vytvárania tohto dokumentu je opísanie fungovania a štruktúry nášho produktu ako aj jeho častí pre potreby ďalšieho vývoja. Okrem implementovaných častí systému sú tu opísané aj moduly či funkcie, ktoré sa ešte len analyzujú a navrhujú, alebo ktorých implementácia je odložená na neskôr.

1.1 Podiel členov tímu na dokumentácií

Nasledujúca tabuľka zobrazuje podiel členov tímu na tvorbe tohto dokumentu.

Úvod	Vladimír Osvald
Ciele projektu na zimný semester	Pavol Michálek
Technologický základ a architektúra	Michal Cihák
Dátový model	Matúš Pikuliak
Zdielanie stretnutia	Michal Gajdoš
Generovanie odkazov	Tomáš Sýkora
Vytváranie stretnutí	Matúš Pikuliak
Časy stretnutí	Lukáš Masár
Miesta stretnutí	Michal Cihák
Hlasovanie	Pavol Michálek
Používatelia	Tomáš Sýkora
Notifikácie	Michal Gajdoš
Módy	Vladimír Osvald

Tabuľka 1. Podiel členov tímu na dokumentácií

2 Ciele projektu pre zimný semester

Cieľom nášho projektu je vytvoriť webovú službu, ktorá používateľom umožní čo najjednoduchšie plánovanie stretnutí za pomoci atraktívneho grafického používateľského rozhrania a bez nutnosti registrácie. Na prihlásenie do služby môžu byť použité existujúce účty zo služieb ako sú napríklad Google či Facebook. Plánovanie stretnutia bude veľmi flexibilné. Zakladateľ udalosti môže definovať miesto a termín stretnutia, vytvoriť hlasovanie za miesto a termín alebo jednoducho nechá pozvaných, aby navrhli čo vyhovuje im. Termín udalosti môže byť konkrétny čas alebo časový úsek. Samozrejmosťou je aj možnosť zahlasovať za viac miest či termínov, pričom bude jasne viditeľné kto hlasoval za čo.

Za zimný semester chceme vytvoriť funkčný prototyp aplikácie, naplňajúci dohodnutú špecifikáciu, ktorý sa pridrža základných hodnôt, ktoré sme si špecifikovali na začiatku práce na projekte. Medzi tieto patrí:

- intuitívnosť a prívetivosť používateľského rozhrania
- jednoduchosť dohadovania stretnutia z hľadiska používateľa
- možnosť upravovať parametre stretnutia podľa svojich požiadaviek

Základná funkcionálnosť, ktorú sme si vytýčili pred začiatkom práce ako súčasť tzv. Minimal value product zahŕňa:

- možnosť práce so stretnutím bez registrácie či iných zdĺhavých procedúr
- možnosť jednoduchého zdieľania stretnutia
- možnosť hlasovania za jednotlivé aspekty stretnutia
- možnosť pridávania možných miest stretnutí s integráciou na Google Maps
- možnosť jednoduchého a rýchleho pridávania časov stretnutí
- zobrazenie stavu stretnutia a hlasovania

3 Celkový pohľad na systém

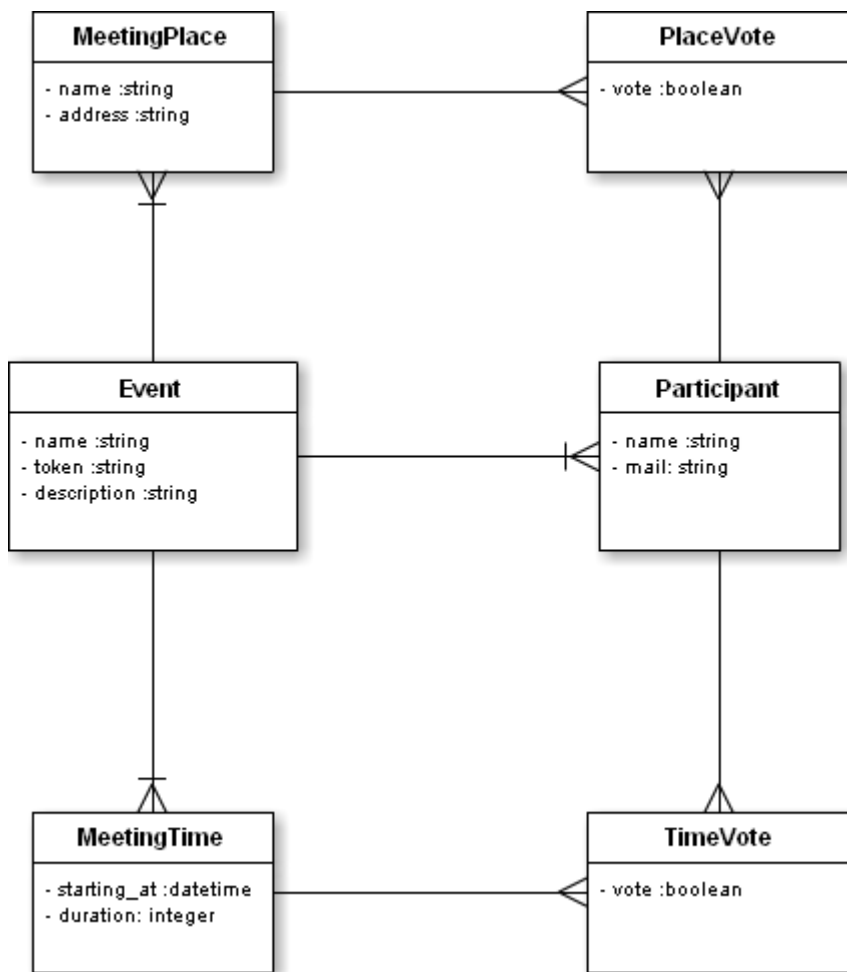
3.1 Technologický základ a architektúra

Náš projekt je webová aplikácia postavená na frameworku *Ruby on Rails*. Táto back-end technológia nám poslúžila aj ako základ pre našu architektúru. Komunikácia s klientom je prirodzene založená na modeli *klient-server*. Rails ako framework konkrétne podporuje vzor

REST ako základ pre prenos dát medzi používateľom a našou aplikáciou. Rails nám takisto poskytuje aj základnú architektúru pre fungovanie samotnej aplikácie. Táto architektúra realizuje známy vzor *Model-View-Controller*. V *Modeli* využívame bežnú knižnicu *Active Record*. Front-end vyvíjame v jazyku *Javascript* s rozšírením o knižnicu *jQuery*. Ako DBMS používame *PostgreSQL*.

Architektúra MVC ako aj knižnica *Active Record* je prístupná v dokumentácii frameworku *Ruby-on-Rails*.

3.2 Dátový model



Obr. 1 - Dátový model systému.

Každá entita má okrem tých vymenovaných ešte atribúty *created_at* a *updated_at*. Základnou dátovou entitou v našom systéme je *Event*, reprezentujúci jednotlivé stretnutia. K jednému eventu prislúcha nula alebo viacero účastníkov (*Participant*). Okrem toho stretnutie má nula alebo viacero možných časov stretnutí a miest stretnutí (*MeetingTime*, *MeetingPlace*). O týchto zasa účastníci dokážu hlasovať, pričom hlasy vyjadrujú príslušné entity *TimeVote* a *PlaceVote*.

Atribút *token* v triede *Event* je unikátny token daného stretnutia, pomocou ktorých sa dá event identifikovať a ktorý sa používa pri vytváraní URL odkazov. Názvy ostatných atribútov sú samovysvetľujúce. Dátové typy sú prebraté z knižnice *Active Record*.

3.3 Moduly

Opísať dizajn obrazoviek a prípadne identifikovať moduly ktoré sa na nich nachádzajú aj s vysvetlením prečo tam sú a ako fungujú, ako spolu súvisia.

3.4 Referencie a dokumenty

- Dokumentácia frameworku Ruby on Rails: <http://rubyonrails.org/documentation/>
- Dokumentácia knižnice jQuery: <http://api.jquery.com/>
- Dokumentácia knižnice Google Maps: <https://developers.google.com/maps/>

4 Moduly

4.1 Zdieľanie stretnutia (čiastočne implementované)

4.1.1 Analýza

Prvou, jednoduchšou možnosťou zdieľania stretnutia je previazanie URL odkazu na konkrétne stretnutie. Takto dokážu používatelia zdieľať stretnutia jednoduchým skopírovaním odkazu a zaslaním svojim priateľom, s ktorými chcú stretnutie zdieľať.

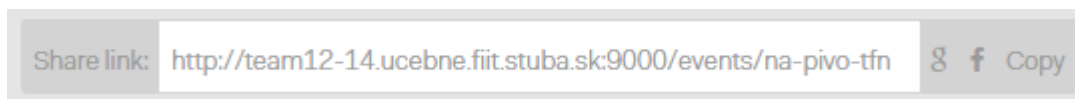
Manuálne zasielanie je funkčné riešenie, je však možné rozšíriť ho integráciou na sociálne či mailové služby. Takáto integrácia by čiastočne zautomatizovala odosielanie odkazov priateľom. V rámci takejto integrácie by si používatelia budovali na stránke vlastný zoznam kontaktov.

4.1.2 Návrh

Pre potreby nášho prototypu sme sa rozhodli nechať zdieľanie na používateľa. Každé stretnutie má unikátny URL odkaz, ktorý sa vygeneruje pri jeho vytváraní. Tento odkaz používateľia môžu zaslať svojim priateľom a tí sa tak dostanú k tomuto stretnutiu. Pre zvýraznenie tejto funkcionality sme sa rozhodli do vrchnej časti stránky stretnutia umiestniť panel s týmto vygenerovaným odkazom a tlačidlo, ktoré tento link nakopíruje do *clipboardu*.

4.1.3 Implementácia

Pri implementácii sme použili gem ZeroClipboard. Zdrojový kód modulu sa nachádza v súbore `/app/views/events/show.html.erb`. Dizajn tohto modelu sa nachádza na Obr. X.



Obr. 2: Zobrazenie modulu zdieľania odkazov

4.2 Generovanie odkazov (implementované)

4.2.1 Analýza

Rozhodli sme sa stretnutia zdieľať pomocou odkazov a preto bolo potrebné vytvoriť metódu, akou ich budeme generovať. Prirodzenou požiadavkou je, aby každé stretnutie malo unikátny odkaz. Ďalšou požiadavkou bolo, aby bol tento odkaz "čitateľný" aj pre človeka, aby už pri zbežnom pohľade používateľ videl, o aké stretnutie sa jedná.

4.2.2 Návrh

Na základe analýzy sme navrhli nasledovný tvar odkazov:

doména_adresa/events/<názov stretnutia>-<token>

, kde názov stretnutia je názov zadaný používateľom očistený o špeciálne znaky a diakritiku a upravený do podoby vhodnej pre URL odkazy a token je unikátna trojica znakov, pomocou ktorých dokážeme rozlíšiť dve stretnutia s rovnakým názvom.

4.2.3 Implementácia

Tokeny sa vytvárajú zároveň so stretnutiami vo funkcii `create_token` v modeli stretnutia v súbore `/app/models/event.rb`. Naviazanie tokenu na URL odkaz má na starosti priamo systém routingu frameworku Rails.

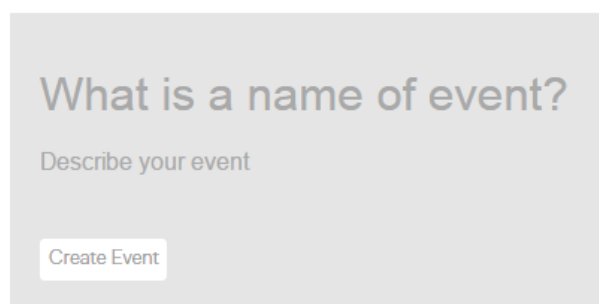
4.3 Vytváranie stretnutí (implementované)

4.3.1 Analýza a návrh

Po preskúšaní viacerých layoutov sme sa rozhodli pre vytváranie stretnutí na úvodnej (landing) stránke nášho produktu. Na tejto stránke uvedieme našu službu a ponúkneme používateľovi jedno pole, názov stretnutia, pomocou ktorého okamžite a rýchlo vytvorí stretnutie, s ktorým potom ďalej dokážeme pracovať. Takáto jednoduchá stránka neodradzuje používateľov zahltením mnohými informáciami.

4.3.2 Implementácia

Obrazok create page - trochu ostrihat našu súčasnu.



Obr. 3: Obrazovka vytvorenia udalosti

4.4 Časy stretnutí (implementované)

4.4.1 Analýza

To, akým spôsobom sa budú zobrazovať a pridávať časy stretnutí bude v konečnom dôsledku závisieť hlavne od frontend-ového zobrazenia týchto časov. Je potrebné aby toto zobrazenie bolo prehľadné a aby sa v ňom ľahko orientovalo, tzn. aby bol používateľ na prvý pohľad schopný

určiť, kedy v čase sa dané stretnutie nachádza. Musí byť zabezpečené čo najjednoduchšie pridávanie nových návrhov časov a jednoduché hlasovanie za navrhované termíny.

4.4.2 Návrh

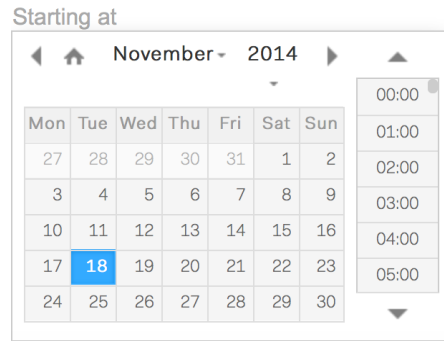
V prvom prototypy sme sa rozhodli zobrazovať tieto časy ako list. Viaceré návrhy časov v tom istom dni budú zobrazené v skupine pod jedným dátumom pre lepšiu prehľadnosť. Každý termín bude zobrazovať presný čas začiatku aj konca stretnutia. Pri každom termíne sa bude nachádzať rýchla možnosť pre hlasovanie a takisto bude zobrazené koľko ľudí už za termín hlasovalo. Pridávanie nových termínov bude umožnené pomocou kalendáru.

4.4.3 Implementácia

Vypísanie všetkých termínov prisluchajucich k danému stretnutiu sa deje v momentálnom prototypy v /app/views/events/show.html.erb. Tento list (obr.) je upravený pomocou CSS. Pri pridávaní nových termínov pomáha jQuery knižnica jquery.datetimepicker.js (obr.). Termíny sú zoradené podľa dátumu a pridanie nového termínu je vždy priamo zapísané do databázy. Pri termínoch sa zvýrazňuje hlasovanie daného používateľa.

11.11. Tuesday	18:00 - 06:00	4		
18.11. Tuesday	18:00 - 06:00	1		
	19:00 - 23:00	0		

Obr. 4: List termínov



Obr. 5: Pridávanie nových termínov

4.5 Miesta stretnutí (implementované)

4.5.1 Analýza

Tento modul musí jasne a rýchlo zobrazit' navrhnuté miesta, dovoliť hlasovanie a pridávanie nových miest. Pre prehľadnosť a zviditeľnenie modulu je vhodné použiť aj mapu. Pridávanie nového miesta musí byť jednoduché, hlavne čo sa týka adries. Preto je vhodné aby adresy boli v reálnom čase ponúkané používateľovi pri písaní.

4.5.2 Návrh

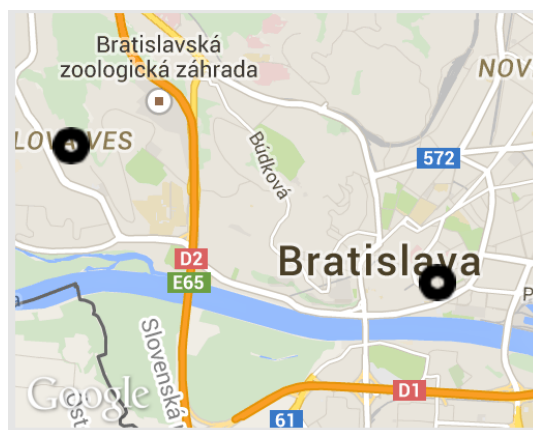
Pre tento prototyp sme sa rozhodli zobrazovať miesta podobne ako časy v liste. Pri každom mieste bude zobrazená aj adresa a následne budú všetky miesta zobrazené aj na mape pre lepší prehľad. Zobrazenie mapy sme zvolili pomocou Google Maps API, čo takisto poskytne aj ponúkanie adries používateľovi.

4.5.3 Implementácia

Vypísanie všetkých miest (obr. X) prisluchajucich k danému stretnutiu sa deje v momentálnom prototypu v `/app/views/events/show.html.erb`. Jednotlivé miesta sú takisto zobrazené v mape (obr. X) pomocou jQuery (main.js) a Google Maps Api. Pridávanie nových miest ponúka adresy (obr. X).

Avion Shopping Park	Avion Shopping Park, 18 Ivanska Cesta Street, 821 04 Bratislava, Slovakia	0		
Fox Valley 1	Pod Rovnicami 41-3, Karlova Ves, Bratislava	5		

Obr. 6: List miest



Obr. 7: Mapa s miestami

Avion|

- Avion Shopping Park Bratislava, Slovakia
- Avion Ivanská cesta, Bratislava-Ružinov, Slo...
- Avion - IKEA Bratislava-Ružinov, Slovakia
- Avion Shopping Park Bratislava-Ružinov,...
- Avion France

powered by

Obr. 8: Ponúkание adries

4.6 Hlasovanie (čiastočne implementované)

4.6.1 Analýza

Hlasovanie a jeho výsledky sú základným výstupom aplikácie. Má jasne ukázať ako prebehla komunikácia medzi účastníkmi, a aká je ich dohoda. Prehľadným spôsobom sa môžu účastníci vyjadriť k navrhnutým časom a miestam.

4.6.2 Návrh

Pri každej navrhnutej položke umožníme účastníkom vyjadriť sa za alebo proti návrhu. Zobrazíme súčet kladných a záporných hlasov. Po prejedení ponad meno každého účastníka uvidíme v zoznamoch časov a miest, za ktorý hlasoval kladne a za ktoré záporne. Všetci účastníci tak získajú kompletný obraz o dostupnosti jednotlivých ľudí, ich preferenciách a najvhodnejších návrhoch.

4.6.3 Implementácia

Komplexné dáta o hlasoch jednotlivých účastníkov vo formáte JSON sú zaslané pri načítaní webovej stránky. Za pomoci frontend-ovej javascript-ovej knižnice jQuery sa používateľom dynamicky zobrazujú hlasy jednotlivých účastníkov. Pri hlasovaní sa odpáli asynchrónne volanie na server. Po spracovaní nového hlasu databázou nasleduje odpoveď v rovnakom formáte ako pri načítaní stránky, a aktualizované informácie sa prepíšu vo webovom prehliadači bez znovunačítania stránky.



Obr. 9: Zobrazenie hlasovania v prehliadači

4.7 Používatelia (čiastočne implementované)

4.7.1 Analýza

Typ produktu na ktorom pracujeme predurčuje dôležitosť modulu používateľov. Na stránku udalosti bude pristupovať viacero ľudí a ich úlohou bude hlasovať za jednotlivé možnosti, prípadne dopĺňať informácie. Používatelia však časom môžu zmeniť svoj názor, a budú chcieť svoje rozhodnutie upraviť. Preto je potrebné jednotlivých používateľov rozlišovať.

Ďalšou úlohou sú role jednotlivých používateľov, nakoľko kompetencie jednotlivých používateľov nebudú rovnaké.

4.7.2 Návrh

Pre účely prototypu bude identifikácia používateľov jednoduchá. Budeme predpokladať, že všetci používatelia budú pracovať len pod svojim menom, ktoré si vyberú zo zoznamu, do ktorého sa budú mať možnosť dopísať, v prípade prvého prístupu na stránku. V neskorších fázach bude potrebné autorizáciu zabezpečiť. Možnosťami pritom sú:

- vlastný účet,
- integrácia s účtami na sociálnych sieťach,
- unikátny prístupový link,
- heslo (zadané pri prvom hlasovaní).

Zároveň bude potrebná implementácia jednotlivých rolí, aby sa tak mohli rozdeliť kompetencie jednotlivých účastníkov. Pre potreby nášho projektu by bolo vhodné implementovať nasledujúce role:

- zakladateľ udalosti - vytvára udalosť, môže pozývať ľudí, nastavuje povolenia pre ostatných používateľov
- admin - takmer rovnaké práva ako zakladateľ
- “môže editovať” - používateľ, ktorý môže pridávať informácie, časy, miesta,...
- “môže hlasovať” - používateľ, ktorý môže za jednotlivé ponúkané možnosti hlasovať
- pozorovateľ - používateľ, ktorý môže len prezerať stav udalosti

Jednotlivé role bude potrebné ešte dôkladnejšie analyzovať, aby sme získali odpovede na nasledujúce otázky:

- Sú potrebné ešte nejaké role?
- Kto môže upravovať práva?
- Bude možné upravovať práva pre jednotlivé role?

4.7.3 Implementácia

Na základe návrhu bola v prototyp implementovaná jednoduchá identifikácia používateľa pomocou označenia svojho mena v zozname účastníkov, ako je vidieť na Obr. 10. Takto označený účastník sa uloží do cookies vo webovom prehliadači pomocou jQuery, na základe čoho sú zobrazené jemu prisluchajúce informácie.



Obr. 10: Zoznam účastníkov

4.8 Notifikácie (neimplementované)

4.8.1 Analýza

Dôležité je určiť, akými kanálmi môžeme používateľa notifikovať a tiež stanoviť, aké notifikácie sú pre používateľa dôležité.

Notifikačné kanály budú kopírovať kanály, ktorými boli jednotlivý používatelia prizvaní do aplikácie na konkrétnu udalosť. Používateľ bude mať tiež možnosť pri prvej identifikácii zadať svoj e-mail, a možnosť zvoliť si ho ako notifikačný kanál.

4.8.2 Návrh

Notifikačný modul bude generickou metódou pre všetky typy notifikácií. Externé spúšťače vyvolajú túto metódu a poskytnú vstupné parametre. Musí mať pevne stanovené rozhranie pre vstupné parametre, aby mohla byť volaná z rôznych častí aplikácie a to aj asynchrónne.

4.9 Módy (neimplementované)

4.9.1 Analýza

Pre ďalšie rozširovanie aplikácie a jej komerčné uplatnenie poskytneme možnosť zvoliť si mód, v akom sa bude vytvorená udalosť zobrazovať účastníkom. Dá sa tak zabezpečiť, aby aplikácia zostala jednoduchá (čo je jej primárnou požiadavkou), a zároveň ponúkala na požiadanie rozšírené možnosti.

4.9.2 Návrh

Pri vytváraní udalosti si používateľ môže zvoliť, že jeho udalosť bude zobrazovaná s rozšírenými možnosťami a bohatším obsahom. Zmena sa prejaví najmä v zobrazení kalendára, a dovoľí používať aplikáciu určenú pre veľký počet účastníkov a na dennej báze udalostí.